



Dostępna pamięć: 128MB

Podniebne miasto

Nie wszyscy wiedzą, bo i nie wszyscy wiedzieć mogą, że wysoko wśród chmur znajduje się podniebne miasto, które składa się z tysięcy unoszących się wysepek, ponumerowanych od 1 do n . Na wyspie o numerze 1 znajduje się karczma, w której do niedawna przysypiał zły prorok. Przydomek „zły” zawdzięcza swojej olbrzymiej niesolidności, która i tym razem daje o sobie znać, gdy nerwowo przegląda rozkład lotów sterowców – jedynego środka komunikacji w podniebnym mieście. Prorok już za chwilę ma wystąpić na wiecu na cześć Wielkiego Wodza. Zgromadzenie odbędzie się na wyspie nr n , więc nie jest pewne, czy uda mu się dotrzeć na czas.

Jako gorliwy opiekun ładu i porządku w Państwie Doskonałym, masz za zadanie sprawdzić, czy zły prorok ma szansę dotrzeć do celu na czas. Jeśli nie ma szans, należy aresztować go za zniewagę majestatu jeszcze przed dotarciem na zgromadzenie.

Korzystając z wszechobecnych kamer i zasady braku prywatności, wiesz że prorok opuścił karczmę w chwili 0. Znasz też rozkład lotów wszystkich sterowców, które są zawsze punktualne (jak na antyutopię przystało). Napisz szybko program, który sprawdzi ile czasu zabierze prorokowi dotarcie na wysepkę, na której odbywa się wiec.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($2 \leq n \leq 300\,000$), oznaczająca liczbę wysepek w mieście. Następnie opisane są odloty sterowców z n kolejnych wysepek (poczynając od pierwszej). Każdy taki opis zaczyna się od wiersza zawierającego jedną liczbę całkowitą m_i ($0 \leq m_i$) – liczbę kursów odlatających z i -tej wysepki. Każdy z następnych m_i wierszy opisu zawiera cztery liczby całkowite $w_{i,j}$, $d_{i,j}$, $t_{i,j}$, $o_{i,j}$ ($1 \leq w_{i,j} \leq n$, $1 \leq d_{i,j} \leq 10^6$, $0 \leq t_{i,j} < o_{i,j} \leq 10^6$), opisujące jeden z kursów i oznaczające odpowiednio: $w_{i,j}$ – numer wysepki, do której leci sterowiec, $d_{i,j}$ – czas lotu sterowca, $t_{i,j}$ – moment, w którym pierwszy sterowiec rozpoczął kurs, $o_{i,j}$ – co ile taki kurs się rozpoczyna.

Jeśli prorok dotrze na wysepkę dokładnie w momencie odlotu sterowca, to będzie mógł na niego wsiąść. To się również tyczy pierwszego odlotu, gdy prorok opuszcza karczmę.

W testach wartych przynajmniej 20% punktów zachodzi: $n \leq 25$ oraz $\sum_{i=1}^n m_i \leq 100$.

W testach wartych przynajmniej 60% punktów zachodzi: $n \leq 100\,000$ oraz $\sum_{i=1}^n m_i \leq 350\,000$.

We wszystkich testach zachodzi warunek: $\sum_{i=1}^n m_i \leq 1\,000\,000$.

Wyjście

Jeśli z powodów logistycznych prorok nie ma szans dolecieć na wiec, należy wypisać **ARESzt**. W przeciwnym razie, należy wypisać jedną liczbę – czas potrzebny prorokowi na dotarcie na wysepkę o numerze n .

Przykład

Wejście	Wyjście
5 2 4 10 0 7 3 1 1 7 0 1 2 7 12 55 2 5 3 3 10 5 2 1 100 0	16



Wejście	Wyjście
3 2 2 5 6 7 2 1 7 125 0 0	ARESzt

ocen1: $n = 250$ i $\sum_{i=1}^n m_i = 62001$, klika bez lotów z ostatniej wysepki, małe losowe czasy

ocen2: $n = 100\,000$, i $\sum_{i=1}^n m_i = 199998$, z i -tej wysepki ($i \neq n$) można szybko przejść do tej o numerze $i + 1$, oraz do n -tej (czasem proporcjonalnym do odległości indeksów)